CLAIMS

What is claimed is:

1. A process for processing an executable embedded software code, said process comprising the steps of:

reading an executable embedded code for one given processor;

extracting code sections from said executable embedded code;

reading a file containing a description of a set of instructions for said given processor, based on concepts of TOKEN, FIELDS, ATTRIBUTES and CONSTRUCTORS of a SLED language, enriched with an additional CLASS definition grouping different instructions under a common label; and

using said description to derive from said TOKEN, FIELDS, ATTRIBUTES, CONSTRUCTORS and CLASS an internal representation taking a form of a decision tree.

2. The process according to claim 1 wherein said decision tree is followed by an algorithm to derive a disassembled program corresponding to said executable embedded code.

3. The process according to claim 2 wherein each piece of embedded code is processed as follows:

starting from a root node and ending to a leaf node of said decision tree;

determining a unique path comprising true branches, having values corresponding to contents of said code sections, ending to a leaf node and executing all tests within said path to identify an instruction corresponding to said section of embedded software code; and

repeating the preceding step until all the embedded code is processed.

4. The process according to claim 1 wherein said internal representation is used for automatically retargeting a Static Code Analyzer which comprises the following steps:

following recognition of an instruction, when a leaf node is reached in a tree visit, such recognized instruction is marked with a class that is found in the leaf node;

exploiting a sequence of recognized instructions with their class label in order to recover a global control-flow graph of the executable embedded code; and

discovering procedures boundaries of the executable embedded code by using the recovered global control flow graph.

5. A method, comprising:

reading a first file having an image of executable binary code for a processor;

extracting code sections forming parts of the executable binary code from the image;

reading a second file having a description of a set of instructions for the processor;

using the description to derive a decision tree representation; and

for each code section, progressing through the decision tree representation to derive at least a portion of a disassembled program corresponding to the executable binary code, until substantially all code sections are disassembled.

6. The method of claim 5 wherein the description of the set of instructions for the processor is based on Token, Fields, Attributes, Constructors, and Class concepts from Specification Language for Encoding Decoding (SLED).

7. The method of claim 5 wherein progressing through the decision tree representation for each code section includes:

starting at a root node of the decision tree representation, determining a path including branches having values corresponding to contents of the code section, until ending at a leaf node;

executing tests within the path to identify an instruction, at the leaf node, corresponding to the code section; and

repeating the determining a path in the decision tree and executing tests therein for each of the other code sections.

8. The method of claim 7, further comprising:

selecting a branch if a test in that branch yields a true result, and then progressing to a next node; and

otherwise not selecting a branch if a test in that branch yields a false result, and then progressing to another branch.

9. The method of claim 7, further comprising:

after identifying an instruction when a leaf node is reached, marking that instruction with a class label that is present at that leaf node;

recovering a global control flow graph of the executable binary code by using a sequence of identified instructions with their class labels; and

using the recovered global control flow graph to determine procedures boundaries of the executable binary code.

10. An article of manufacture, comprising:

a machine-readable medium having instructions stored thereon to:

read a first file having an image of executable binary code for a processor;

extract code sections forming parts of the executable binary code from the image;

read a second file having a description of a set of instructions for the processor;

use the description to derive a decision tree representation; and

for each code section, progress through the decision tree representation to derive at least a portion of a disassembled program corresponding to the executable binary code, until substantially all code sections are disassembled.

11. The article of manufacture of claim 10 wherein the description of the set of instructions for the processor is based on Token, Fields, Attributes, Constructors, and Class concepts from Specification Language for Encoding Decoding (SLED).

12. The article of manufacture of claim 10 wherein the instructions to progress through the decision tree representation for each code section includes instructions to:

starting at a root node of the decision tree representation, determine a path including branches having values corresponding to contents of the code section, until ending at a leaf node;

executing tests within the path to identify an instruction, at the leaf node, corresponding to the code section; and

repeat the determining a path in the decision tree and executing tests therein for each of the other code sections.

13. The article of manufacture of claim 12 wherein the machine-readable medium further includes instructions stored thereon to:

select a branch if a test in that branch yields a true result, and then progress to a next node; and

otherwise not select a branch if a test in that branch yields a false result, and then progress to another branch.

14. An article of manufacture, comprising:

a machine-readable medium having instructions stored thereon to:

extract code sections forming parts of an executable code;

load a description of a set of instructions;

use the description to derive a decision tree representation; and

for each code section, progress through the decision tree representation to identify an instruction from the set of instructions that corresponds to the code section, until substantially all code sections have been disassembled via identification of each to an instruction.

15. The article of manufacture of claim 14 wherein the description of the set of instructions is based on Token, Fields, Attributes, Constructors, and Class concepts from Specification Language for Encoding Decoding (SLED).

16. The article of manufacture of claim 14 wherein the instructions to progress through the decision tree representation for each code section includes instructions to:

starting at a root node of the decision tree representation, determine a path including branches having values corresponding to contents of the code section, until ending at a leaf node;

execute tests within the path to identify an instruction, at the leaf node, corresponding to the code section; and

repeat the determining a path in the decision tree and executing tests therein for each of the other code sections.

17. The article of manufacture of claim 16 wherein the machine-readable medium further includes instructions stored thereon to:

select a branch if a test in that branch yields a true result, and then progress to a next node; and

otherwise not select a branch if a test in that branch yields a false result, and then progress to another branch.